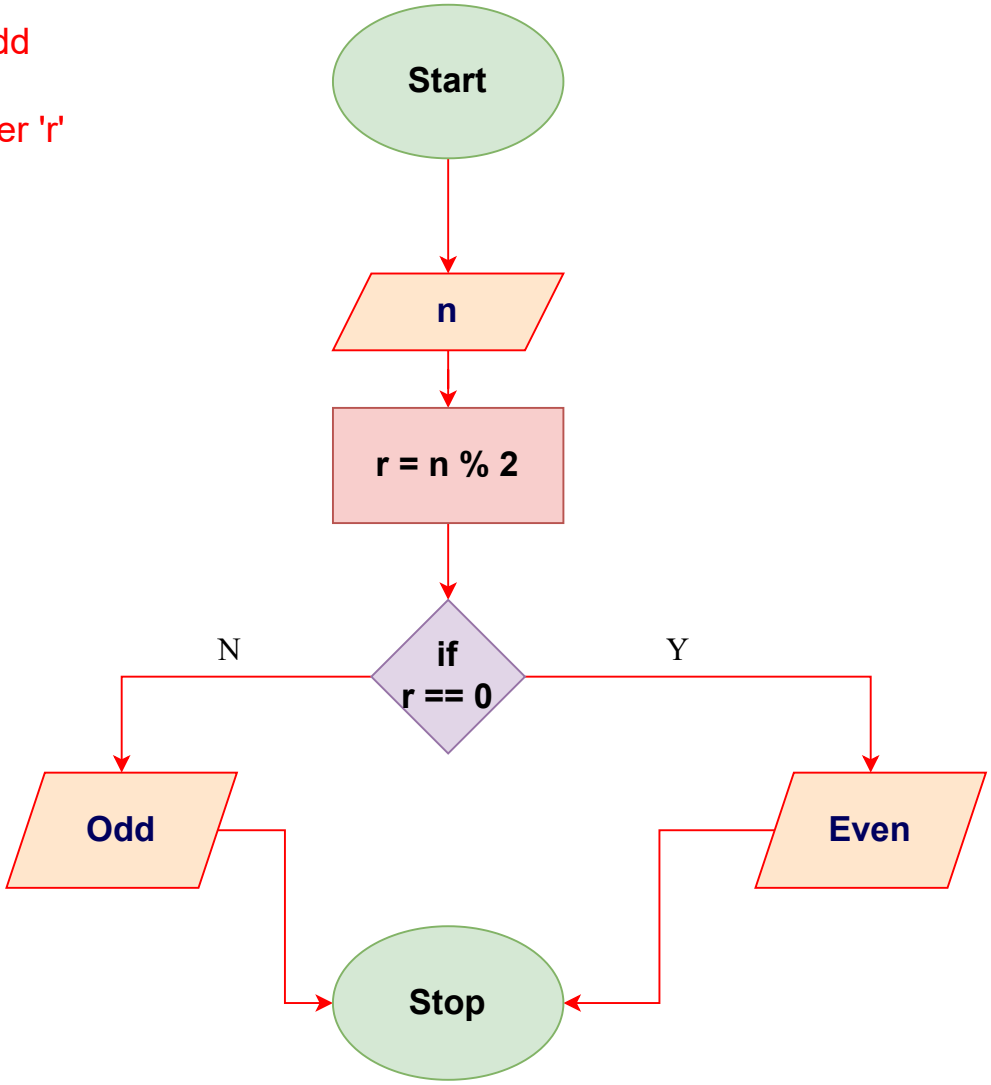
#1
input: number 'n'
output: Even, Odd

process: reminder 'r'



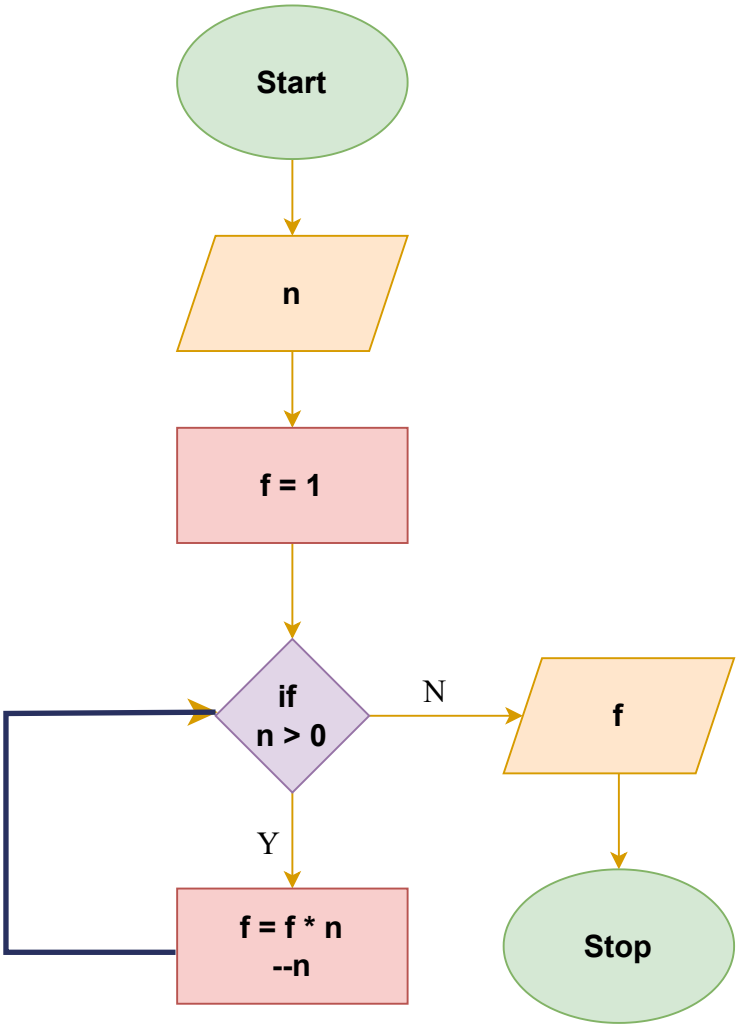
Algo:

1. Enter a number n
2. perform mod operation and find reminder r
3. if r == 0 it is an Even number
   if r != 0 , it is an Odd number

#2
input:- number 'n'

output:- factorial 'f'

process:- loop(for-loop)



Algo:

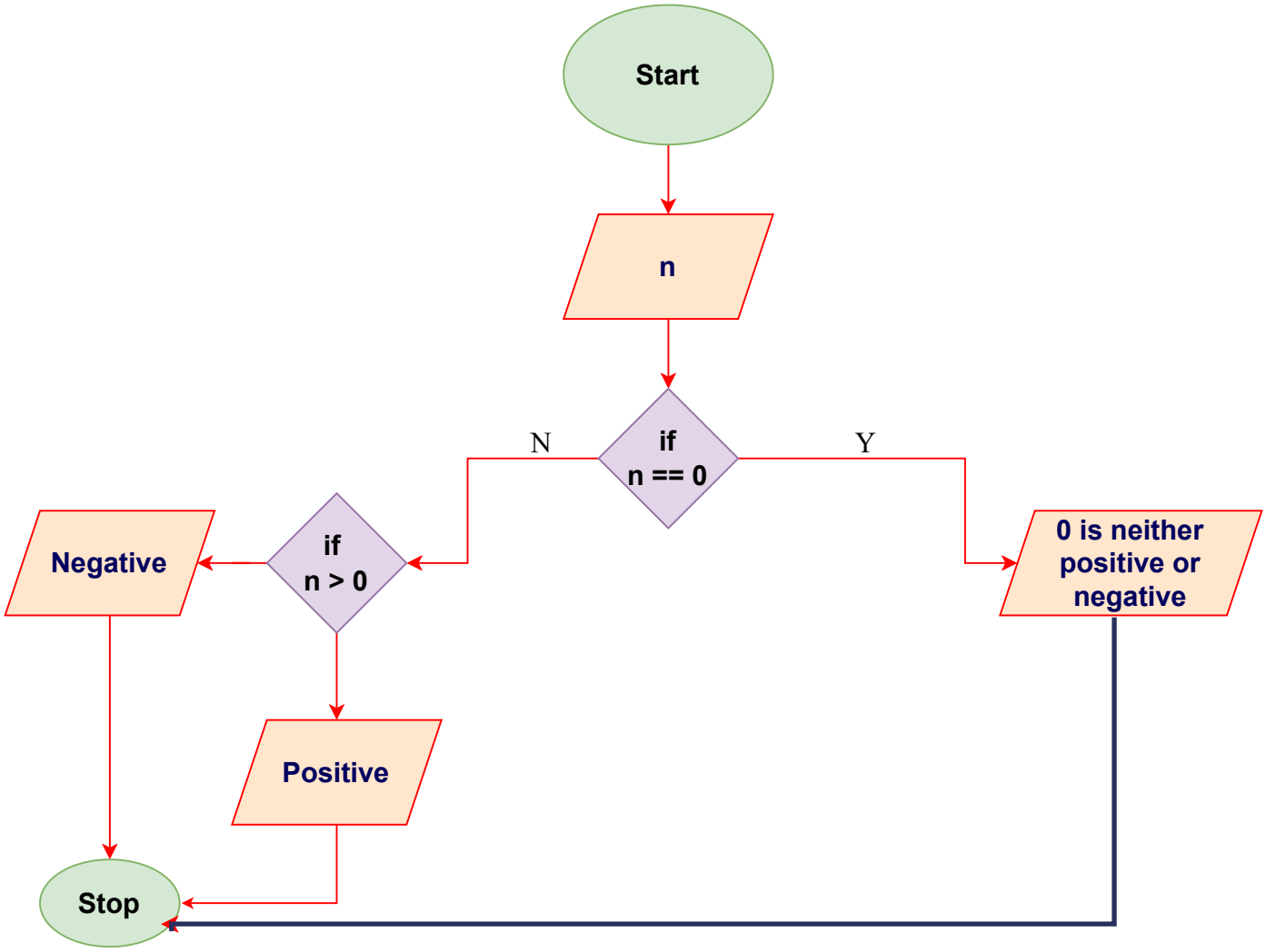
1. Enter a number n whose foctorial is required
2. Set a variable f = 1.
3. Check if n >0.

3a. if True - perform operation f=f*n
decrement n - --n.

3b. if False - print factorial f.

4. Repeat Step 3. untill n>0 false.

#5
input:- number 'n'
output:- Positive, Negative,
Neither positive nor Negative



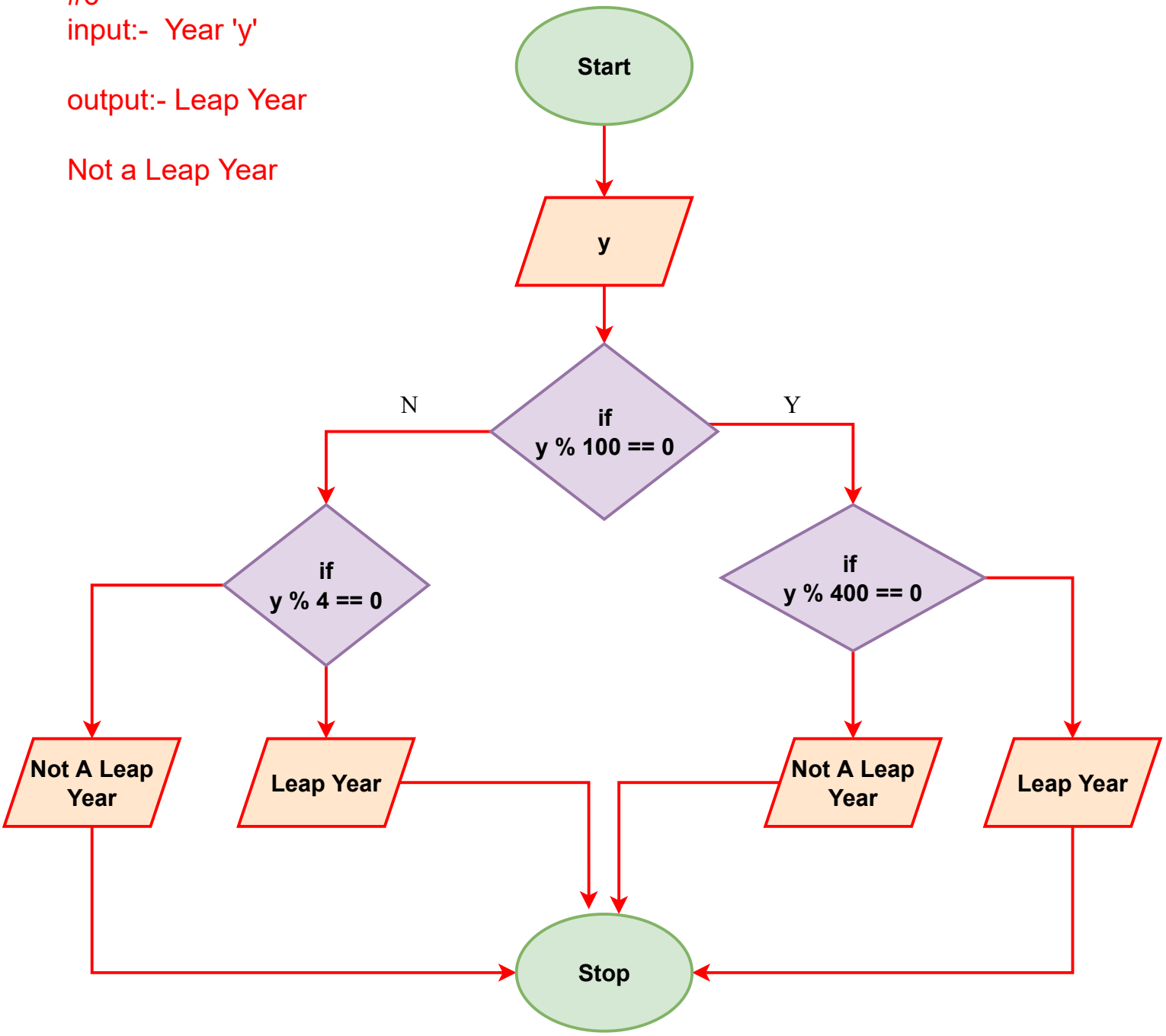
Algo:

1. Enter a number n to check.
2. First check for 0. Apply condition n == 0.
2a. If n == 0 i.e (True)- number is neither negative nor posetive
2b. If n!=0 i.e (False)- check for next condition
2b.1. Apply condition n > 0
2b.1a. If condition true - Number is Positive number
2b.1b. If condition false - Number is Negative number

#6
input:- Year 'y'

output:- Leap Year

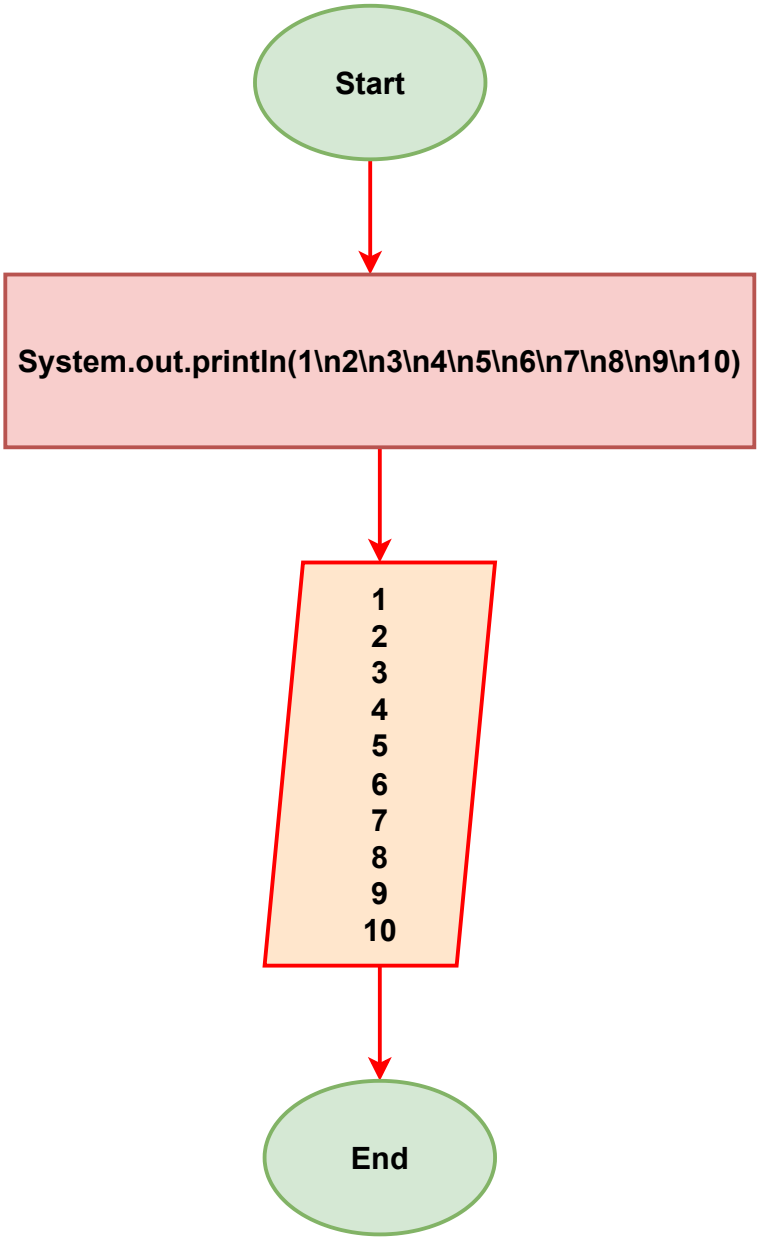
Not a Leap Year



Algo:

1. Enter a Year y to check leap year.
2. First check century year. Apply condition y%100 == 0.
2a. If condition is true - Check Century Leap year. Apply condition y%400 == 0.
2a1. If condition is true - Century year is a Leap Year.
2a2. If condition is false - Century year is not a Leap Year.
2b. If Step 2. condition is false - Check for Normal Leap year. Apply condition y%4 == 0.
2b1. If condition is true. Year is a Leap Year.
2b2. If condition is fale. Year is a Normal Year.

#7
output: 1 to 10

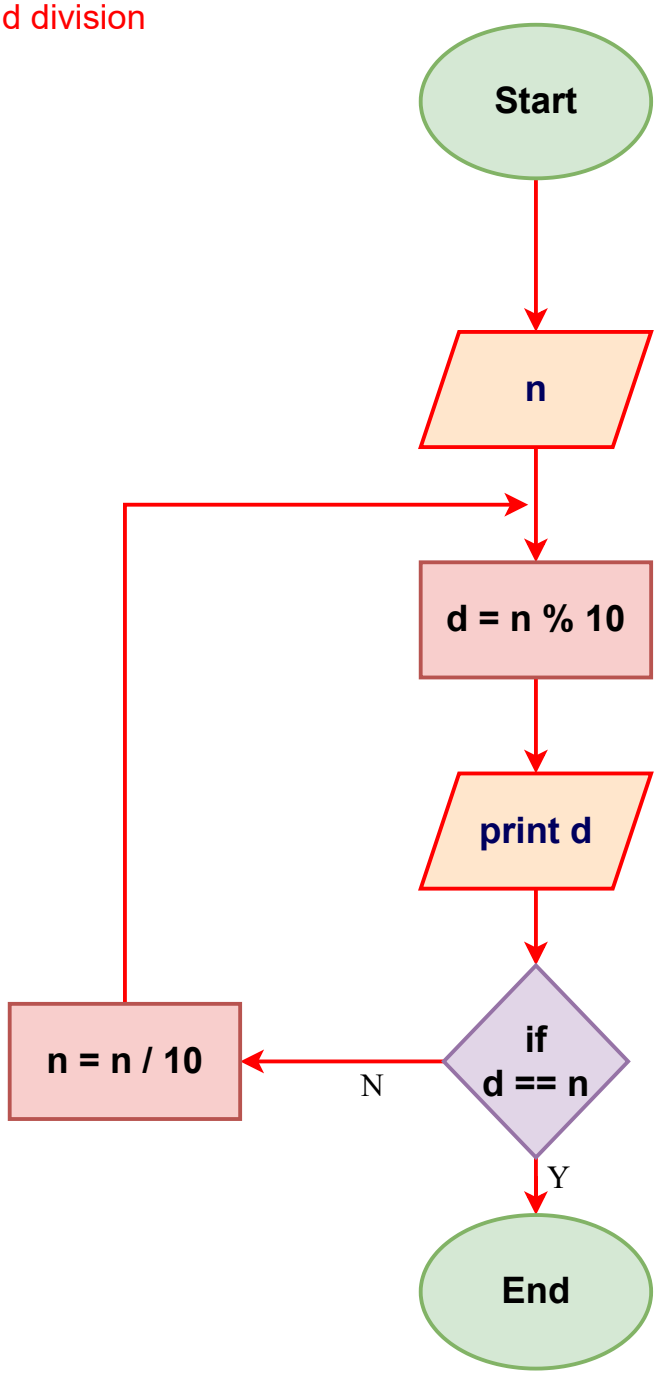


Algo:

1. Print - System.out.println(1\n2\n3\n4\n5\n6\n7\n8\n9\n10);
2. This will print 1-10.

#8
input: number 'n'
output: digits 'd'

process: reminder and division



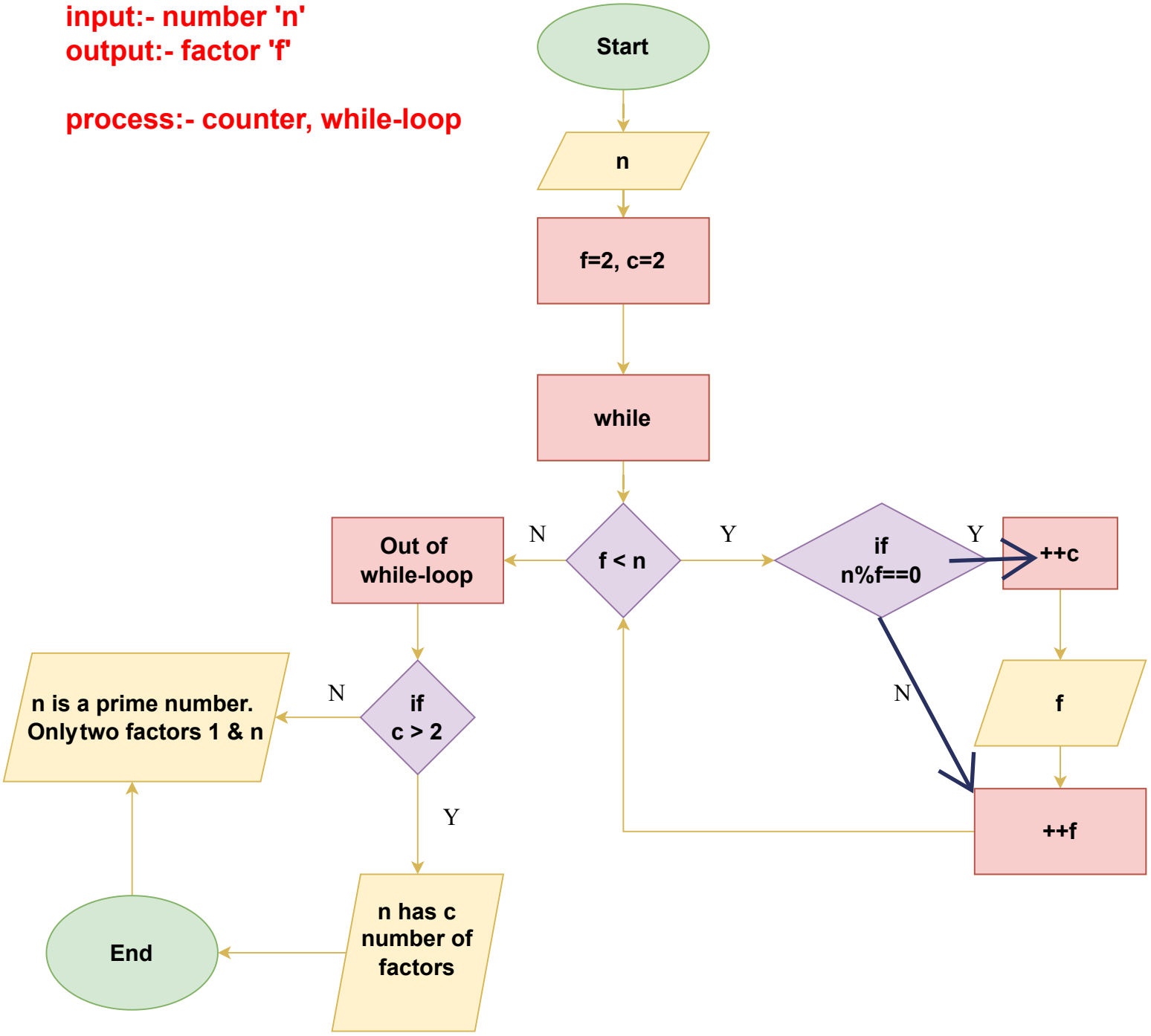
Algo:

1. enter a number n
2. Get unit digit from it. Apply operation n%10 and get reminder 'd'.
3. Print unit digit.
4. Now Apply n/10 and seperate unit digit from it. This operation will convert the number in number with one digit less.
5. Repeat step 2-4 untill d == n.
6. When above conditon satisfy. break the loop to end program.

#9

input:- number 'n'
output:- factor 'f'

process:- counter, while-loop



Algo:

1. Enter an number n to find factors.
2. Set two variables, factor f=2, counter c=2. To find factor from 2 onwards and count number of factors respectively.
3. Create a while loop with condition f<n. To check factors till one less than provided number.

3a. If condition true enter while loop. And start performing mod operation on number with succesive numbers from 2 to n-1. Apply n%f==0.

3a1. if conndition true- increase the counter and increment the value of f. Print factor 'f'.

3a2. if condition fails- just increment value of f.

3a3. Perform step 3 to above untill loop breaks.

3b. when condition at 3a. fails loop get closed.

4. Now check for counter value.

4a. if c==2 i.e n is a prime number.

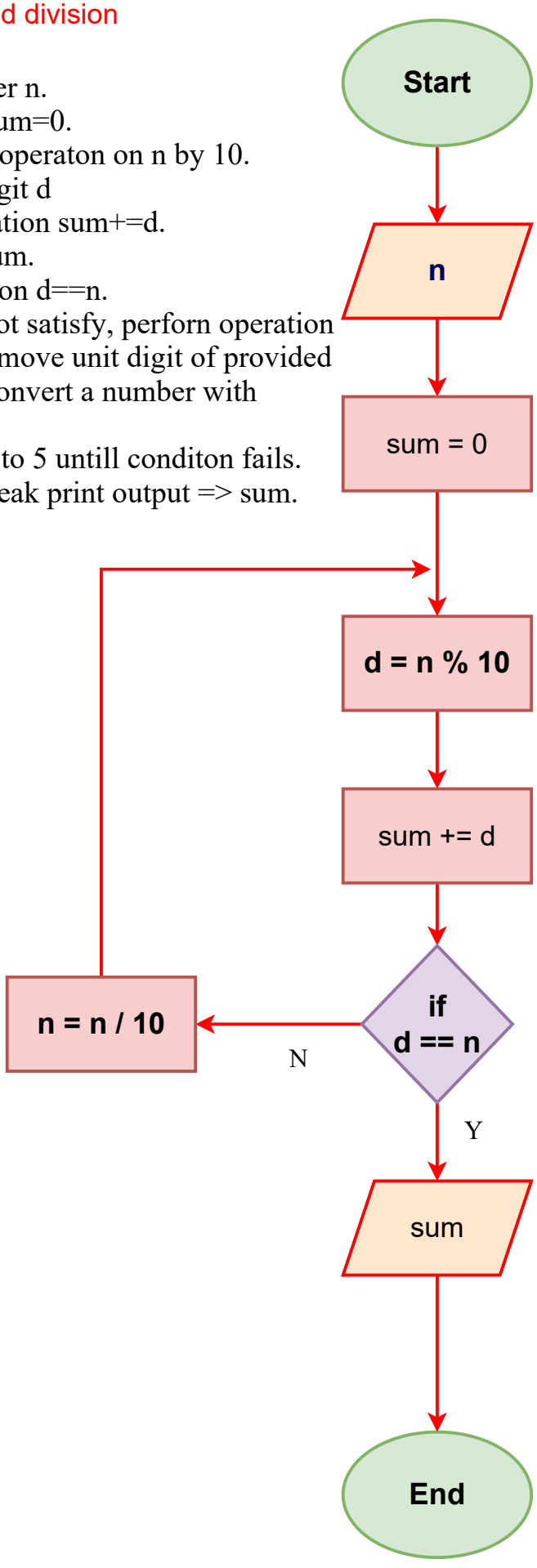
4b. if c>2 i.e n is not a prime number. display number of factors.

#10
input: number 'n'
output: sum of digits 'sum'

process: reminder and division

Algo:

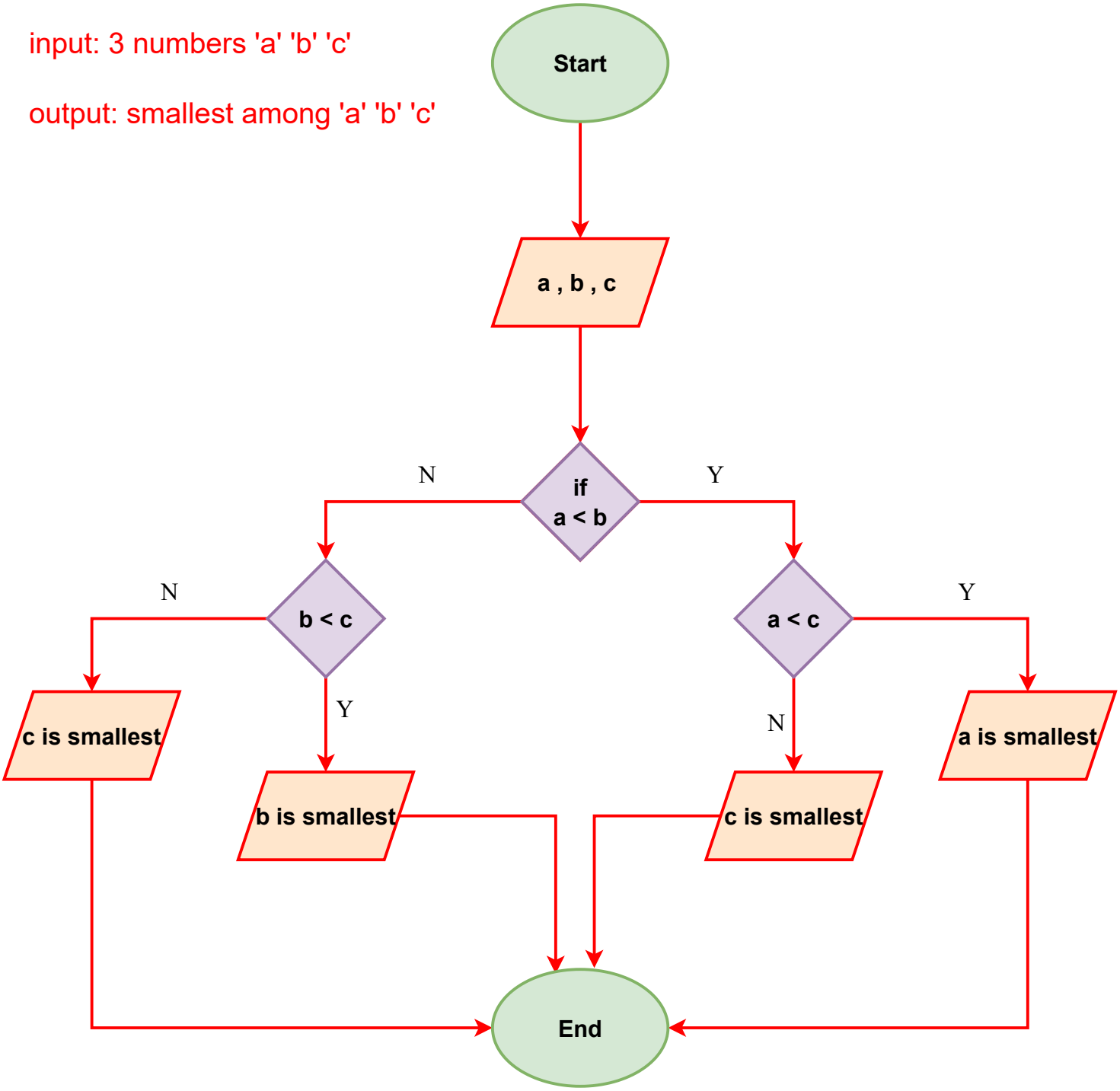
1. Enter a number n.
2. Set variable sum=0.
3. Perform mod operaton on n by 10.
   to find unit digit d
4. perform operation sum+=d.
   to calculate sum.
5. Apply condition d==n.
6. If condition not satisfy, perform operation
   n=n/10. To remove unit digit of provided
   number and convert a number with
   one digit less.
7. Repeat step 3 to 5 untill conditon fails.
8. When loop break print output => sum.



#11

input: 3 numbers 'a' 'b' 'c'

output: smallest among 'a' 'b' 'c'



Algo:

1. Enter three numbers. a,b,c
2. compare a&b. Apply condition- a<b

2a. If a is smaller than b - compare a&c.

2a1. If a is smaller than c- a is smallest number.

2a2. else c is smallest.

3. if a>b, compare b&c.

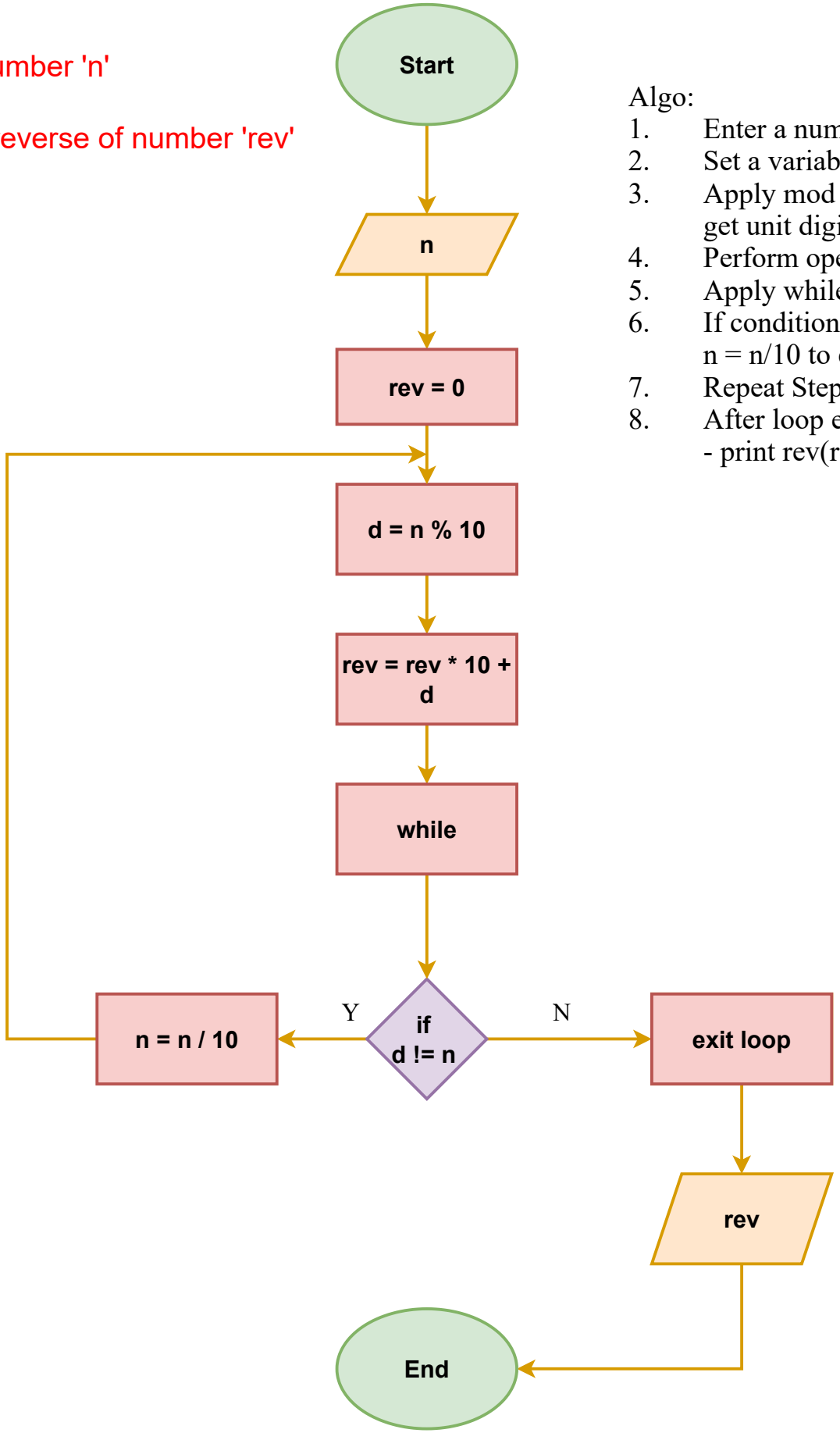
3a. if b<c, b is smallest

3b. else c is smallest.

#13

input: number 'n'

output: reverse of number 'rev'



Algo:

1. Enter a number
2. Set a variable rev=0
3. Apply mod operation on n to get unit digit d.
4. Perform operation rev=rev*10+d
5. Apply while condition- d!=n.
6. If condition true- perform: n = n/10 to eleminate unit digit.
7. Repeat Step 3-6 untill while fails.
8. After loop ends - print rev(result).

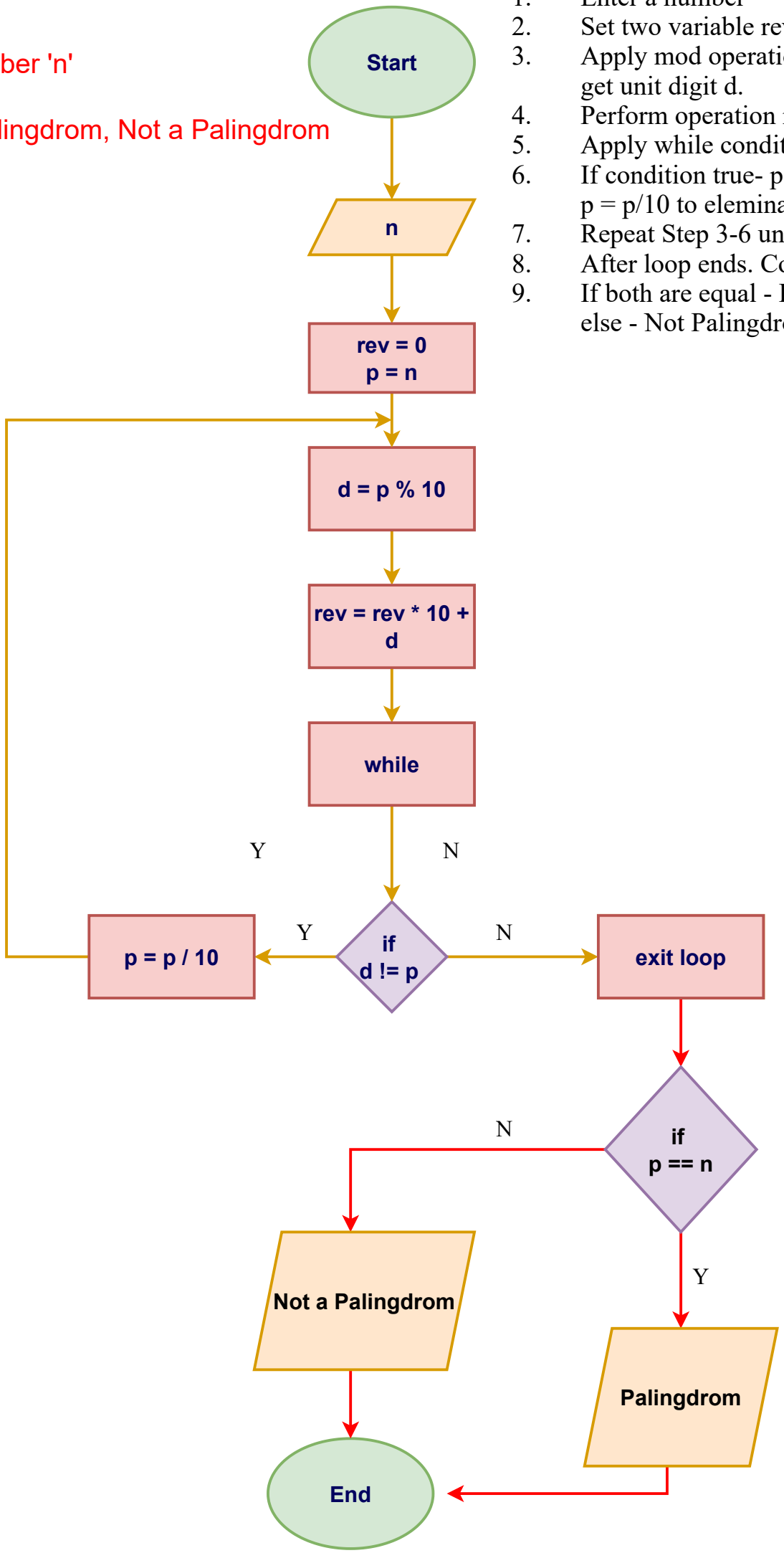
#17

input: number 'n'

output: Palingdrom, Not a Palingdrom

Algo:

1. Enter a number
2. Set two variable rev=0, p=n
3. Apply mod operation on p to get unit digit d.
4. Perform operation rev=rev*10+d
5. Apply while condition- d!=n.
6. If condition true- perform: p = p/10 to eleminate unit digit.
7. Repeat Step 3-6 untill while fails.
8. After loop ends. Compare p & n.
9. If both are equal - Palingdrome else - Not Palingdrome



#19

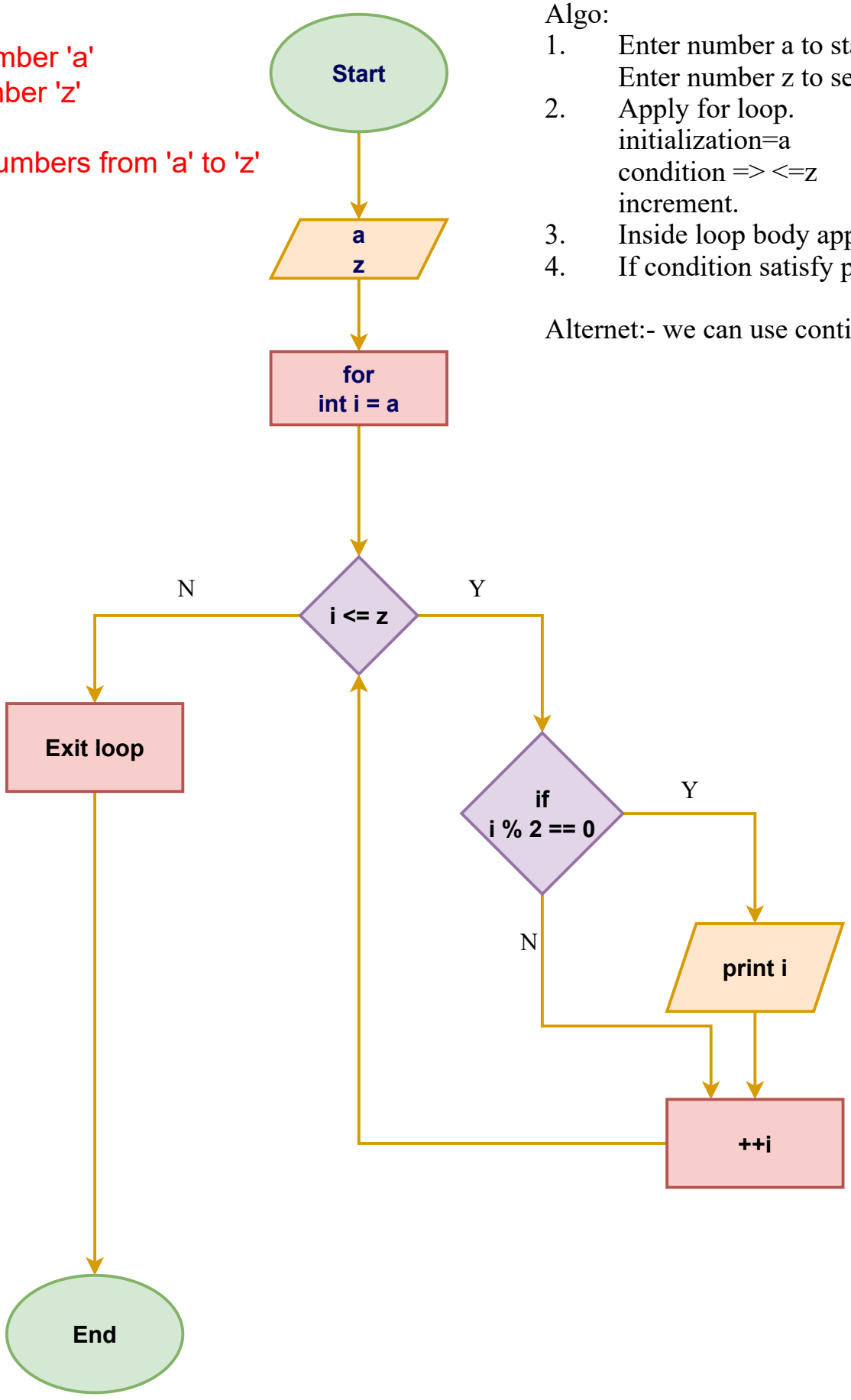
input: initial number 'a'
final number 'z'

output: Even numbers from 'a' to 'z'

Algo:

1. Enter number a to start from.
   Enter number z to search till...
2. Apply for loop.
   initialization=a
   condition => <=z
   increment.
3. Inside loop body apply - i%2 == 0
4. If condition satisfy print i.

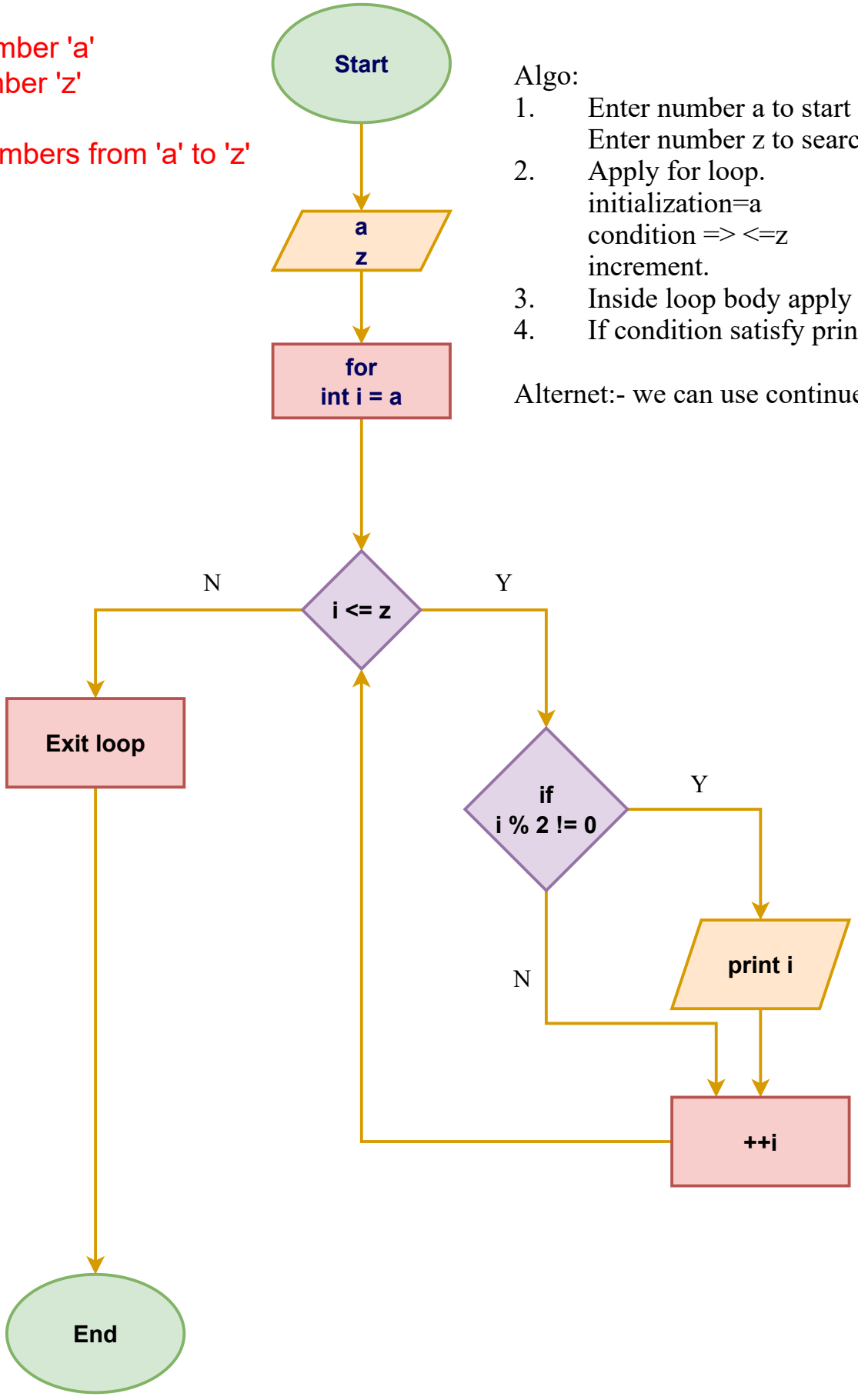
Alternet:- we can use continue keyword



#20

input: initial number 'a'
final number 'z'

output: Odd numbers from 'a' to 'z'



Algo:

1. Enter number a to start from.
   Enter number z to search till...
2. Apply for loop.
   initialization=a
   condition => <=z
   increment.
3. Inside loop body apply - i%2 != 0
4. If condition satisfy print i.

Alternet:- we can use continue keyword